Enabling Access to Applications with Shibboleth:
Adding Shibboleth Logins to the Plone CMS

Alan Brenner

http://tid.ithaka.org/shibplone.pdf
Overview

- Zope and Plone
  - Zope as the Application Server
  - Plone as the Content Management System running in Zope
- End User Login Sequence Overview
- Apache Configuration
  - Apache as the Shibboleth Service Provider
  - Getting data from Apache to Zope
- AutoUserMakerPASPlugin
  - Overview and Installation
  - Configuration options
  - Python Source
- ShibbolethLogin
  - Installation, Configuration and Python Source
- ShibbolethPermissions
  - Installation and Configuration
  - Users grant permissions to their stuff
  - Python Source
Who is Ithaka and Who am I?

• “Ithaka is an independent not-for-profit organization with a mission to accelerate the productive uses of information technologies for the benefit of higher education worldwide.”
  • Research, Strategic Services, Shared Services (IT, HR, etc.)
  • Affiliated with JSTOR and ARTstor
  • “Incubating” Aluka, NITLE and Portico
  • http://www.ithaka.org/

• I am a Senior Engineer in the Technology Innovation and Development group. I develop software, run web sites, databases, and email and VoIP systems, and administer several servers. I have developed these Plone ‘Products’ under the direction of the Research in Information Technology program of the Andrew W. Mellon Foundation.
  • http://tid.ithaka.org/
  • http://rit.mellon.org/
Zope and Plone

• Python
  • Both Zope and Plone are (mostly) written in Python, with a bit of C in Zope for performance in its built-in database
  • http://www.python.org/

• Zope
  • “An open source application server for building content management systems, intranets, portals and custom applications”
  • http://www.zope.org/

• Plone
  • “A ready-to-run content management system”
  • Like Zope, it is extensible with ‘Products’ and Python modules
  • http://plone.org/
  • Currently at 3.0. The Shibboleth integration code is still for Plone 2.5.
Initial Login Sequence

I focus on the Apache to Zope communication in this presentation.
This is the default initial page. ShibbolethLogin modifies the page at the "log in" link in the upper right. You can easily remove the "log in" box ('portlet') on the left.
Plone: Login Sequence #2

ShibbolethLogin adds these ‘Where are you from’ links to Shibboleth Identity Providers. You specify the link titles and targets in the Zope Management Interface.

The “Account details” box encloses the login for local to Plone accounts.
Login Sequence #3

Shibboleth Authentication Request Processed

You are automatically being redirected to the requested site. If the browser appears to be hung up after 15-20 seconds, try reloading the page before contacting the technical support staff in charge of the desired resource or service you are trying to access.

Redirecting to requested site...

Firefox briefly displays the redirect that came from the IdP.
And now I'm logged in.

Note my full name, and the sharing tab. My name is there because Shibboleth provided a header that got saved as my name. ShibbolethPermissions uses sharing tab.
Apache Configuration

- Install mod_shib
- mod_php is helpful during setup.
  - Create an index.php like:
    ```html
    <html><head><title>env</title></head><body><?php phpinfo(); ?></body></html>
    ```
  - This shows all of the values that are (or are not) getting set by mod_shib
- Configure Apache to proxy HTTP connections for Zope
- Configure Apache to rewrite HTTPS connections for Zope
Apache: httpd.conf and modules

- This can go in the httpd.conf:

  ```
  # Load the SHIBBOLETH module
  LoadModule mod_shib /usr/local/shibboleth-sp-1.3/libexec/mod_shib_22.so
  # This is the XML file that contains all the global, non-apache-specific
  # configuration. Look at this file for most of your configuration parameters.
  ShibSchemaDir /usr/local/shibboleth-sp-1.3/share/xml/shibboleth
  ShibConfig /usr/local/shibboleth-sp-1.3/etc/shibboleth/shibboleth.xml
  ```

- These modules also get loaded in the httpd.conf:

  ```
  LoadModule proxy_module modules/mod_proxy.so
  LoadModule proxy_http_module modules/mod_proxy_http.so
  LoadModule rewrite_module modules/mod_rewrite.so
  ```
Apache: HTTP Proxy

• (Almost) Minimal configuration:

```html
<VirtualHost 192.168.191.1:80>
  ServerName alan.ithaka.org
  ServerAdmin alan.brenner@ithaka.org
  DocumentRoot /usr/local/apache-httpd-2.2.4/htdocs
  ProxyRequests Off
  ProxyPass /index.php !
  ProxyPass / http://127.0.0.1:8253/VirtualHostBase/http/alan.ithaka.org:80/test/VirtualHostRoot/
</VirtualHost>
```

• This configures Apache to get pages from localhost port 8253.
• This tells Zope that pages it returns have http://alan.ithaka.org/ as the base URL.
• It also tells Zope to fetch all pages from its /test folder (my Plone install).
• You might want to add Apache caching, other non-proxied URLs, etc.
• This is standard Apache in front of Zope, as documented at Plone’s web site.
Apache: HTTPS Rewriting 1

- Apache listens on 443
  `<VirtualHost 192.168.191.1:443>

- Tell Apache to Proxy everything
  - Sort of, since we don’t tell Apache to proxy
  - This gets most of the HTTP Headers passed through to Zope
    `<Proxy *>
    Order deny,allow
    Allow from all
    `</Proxy>`

- Tell Apache to rewrite the header to set X_REMOTE_USER
  - The REMOTE_USER value is looked up
  - The value is stored temporarily
  - The value is saved in the X_REMOTE_USER header
    `RewriteEngine On`
    `RewriteCond %{LA-U:REMOTE_USER} (.+)
    RewriteRule .* - [E=RW_RU:%1]
    RequestHeader set X_REMOTE_USER %{RW_RU}e`
Apache: HTTPS Rewriting 2

- Tell Apache not to rewrite these (my IdP is on the same Apache)
  
  ```
  RewriteCond %{REQUEST_URI} !^/(shibboleth-(sp|idp)|Shibboleth.sso|SAML)
  ```

- Tell Apache to rewrite everything else to Zope
  - Mostly like the HTTP proxy.
  - Add $1 to the end of the Zope URL to pass the rest of the URL.
    ```
    RewriteRule ^/(.*) http://127.0.0.1:8253/VirtualHostBase/https/alan.ithaka.org:443/test/VirtualHostRoot/$1 [L,P]
    ```

- Setup Shibboleth Authentication
  ```
  <Location />
  AuthType shibboleth
  ShibRequireSession Off
  require shibboleth
  </Location>
  </VirtualHost>
  ```
AutoUserMakerPASPlugin Overview

• Overview
  • PAS is the “Pluggable Authentication System” added to Plone 2.5
  • AutoUserMakerPASPlugin allows Zope to delegate authentication
    • Works with any authentication system available to Apache, not just Shibboleth
    • Can use values from Apache for authorization (to set Plone roles) as well

• Origin
  • Originally a Zope/Plone Product called apachepas by Rocky Burt on behalf of Zest Software (http://zestsoftware.nl/)
  • Updated and turned in to AutoMemberMakerPASPlugin by Erik Rose of the WebLion project at Penn State (http://weblion.psu.edu/)

• Installation
  • Standard Plone Product install (same for ShibbolethLogin and ShibbolethPermissions)
  • Unzip in the Plone $INSTANCE_HOME/Products directory
  • Restart Zope
  • Go to the Add/Remove Products page, click the checkbox by the product name and the install button
Plone: Installing the Products

Add/Remove Products

This is the Add-on Products install section, you can add and remove products in the lists below.

To make new products show up here, put them in the directory /usr/local/test-2.9.7-2.5.3/Products on the file system, and restart the server process.

Products available for install

- AutoUserMakerPASPlugin 0.6
  - Product Description

- ShibbolethLogin 0.6
  - Product Description

- ShibbolethPermissions 0.6
  - Product Description

Installed products

- Archetypes 1.4.4-final
  - Product Description
  - Install log

- CMFPlacefulWorkflow 1.0.5
  - Product Description
  - Install log

- CMFSquidTool
  - Install log

- CacheSetup
  - Install log

- Marshall
  - Install log

- MimetypesRegistry 1.5.0-final
Plone: After Installing the Products

Add/Remove Products

This is the Add-on Products install section, you can add and remove products in the lists below.

To make new products show up here, put them in the directory /usr/local/test-2.9.7-2.5.3/Products on the file system, and restart the server process.

Products available for install

Installed products

- Archetypes 1.4.4-final
  - Product Description
  - Install log
- AutoUserMakerPASPlugin 0.6
  - Product Description
  - Install log
- CMFPlacefulWorkflow 1.0.5
  - Product Description
  - Install log
- CMFSquidTool
  - Install log
- CacheSetup
  - Install log
- Marshall
AutoUserMakerPASPlugin Configuration

- Configured through the “Zope Management Interface”, ZMI
- Configure domain name removal (if needed--probably should change)
  - The default is to remove all domain names from user ID’s
  - This can map multiple people to the same Plone account
    (abrenner@ithaka.org and abrenner@internet2.edu both are abrenner to Plone)
- Configure the HTTP headers to look for (if needed)
- Configure the headers to user for authorization (if needed)
  - If you add headers to the the User Mappings field, the AuthZ tab will allow
    you to assign Plone roles and groups, and to map existing Plone users, to
    header values.
  - If you add headers to the User Sharing fields, users can grant permissions
    to people who have not ever logged in (limited to what they have
    permissions on). AutoUserMakerPASPlugin will assign the granted
    permissions to a new user, immediately after it creates the user.
ZMI: acl_users

/test is the Plone install in the Zope environment.
acl_users is the user and group repository.
The AutoUserMakerPASPlugin and other entries are links to the configuration pages.
ZMI: AutoUserMakerPASPlugin

Only use the strip all option when you know there will not be ID duplication.

Use the strip from list option when you already have users in Plone and want to migrate to external authentication.
I recommend a configuration like this, unless you absolutely know you will not have user ID conflicts.
The defaults might be all you need.
You can list sources for each field. AUMPP will use the first non-empty entry in the list.
AutoUserMakerPASPlugin fills these values in, when the User Setup Headers have data. As mentioned in the login sequence, Plone shows the “Full Name” instead of the userid when this has data.
ZMI: AutoUserMakerPASPlugin 3

User Mapping Headers

Use the items below for global authorization/account mapping. **Enter one per line.** Examples include
HTTP_X_REMOTE_USER, HTTP_SHIB_EP_PRINCIPALNAME, HTTP_SHIB_EP_ORGDN, and HTTP_SHIB_EP_ENTITLEMENT. If this maps to more than one user id, then this effectively declares a group.

User Sharing Headers

Make the items below available for user sharing, when ShibbolethPermissions is installed. **Enter one per line.** Use
the item in the second box as the label users see. Labels must be in the same order as the sources. Non-existent
labels will use the source name. Examples are the same as the global authorization values, above.

You only need to fill these in when you, or your users, want to grant permissions to users authenticated via AutoUserMakerPASPlugin.
When you add lines to the User Mapping Headers, AutoUserMakerPASPlugin enables assigning global Plone roles.

Adding User Sharing Headers allows users to grant permissions to users who don't exist yet.
The initial role mapping shows anyone with 'abrenner' in the HTTP_X_REMOTE_USER getting assigned to the Administrators group at first log in.
After clicking the Add button, AutoUserMakerPASPlugin has a mapping, so it shows it with some edit/delete options.
This shows that the external user 'abrenner' will be the Plone user 'me'. Handy for migrating existing Plone sites, so users have one less password to deal with.
AutoUserMakerPASPlugin Source

- **AutoUserMakerPASPlugin/Extensions/Install.py**
  - Instantiates and activates as the Authentication and Extraction interfaces

- **AutoUserMakerPASPlugin/auth.py**
  - Defines the ExtractionPlugin class
    - Implements the IExtractionPlugin interface
    - The extractCredentials method gets the HTTP request values configured by the Plone administrator and stores them for use by a class that implements the IAuthenticationPlugin interface.
  - Defines the AutoUserMakerPASPlugin class
    - Implements the IAuthenticationPlugin interface
    - The authenticateCredentials method creates Plone users from Shibboleth authenticated users, and assigns roles and permissions to those users.

- **Defines the ApacheAuthPluginHandler class**
  - handles the ZMI configuration forms
ShibbolethLogin

- Installing this modifies the default login_form, adding a section that show the configured Identity Providers.
  - This only works when you haven’t modified the default login_form.
  - You can use the tal:block in a customized form:
    
    ```
    <tal:block tal:define="wayf here/acl_users/ShibbolethLogin">
      <p i18n:translate="description_external_login"
        tal:repeat="idp python:wayf.getWayf(came_from)">
        Log in with a <a href="" tal:attributes="href python:idp[1]">PROVIDER</a> user id.
      </p>
    </tal:block>
    ```
  - Configuration is all on one page in the ZMI (acl_users/ShibbolethLogin).
    - The configuration page has 2 defaults as examples.
      - The first example should have all of the ‘localhost’s replaced with your configuration.
      - The second example works as-is: ProtectNetwork’s identity provider.
    - The Plone administrator needs to configure this right after installation.
      - At least remove the “localhost” entries.
ZMI: ShibbolethLogin Configuration

Replace at least all of the 'localhost's with your server.

The big fields are a (meta-) WAYF.
ShibbolethLogin Python

- ShibbolethLogin/Extensions/Install.py
  - Instantiates an instance in the acl_users folder
  - Gets the default login_form, modifies it to add the TAL block, and saves the updated form in portal_skins/custom
  - Installs a custom logout_form that redirects users back to the HTTP site.

- ShibbolethLogin/redirector.py
  - Defines a ShibbolethLogin class
    - retrieves configuration values for the login and logout forms
  - Defines a ShibbolethLoginHandler class
    - handles the ZMI configuration forms
ShibbolethPermissions

• Installing this modifies the default Plone sharing page, adding a section for granting permissions to new users based on incoming attributes.
  • Already existing Plone users, regardless of how they authenticate, are handled by the existing Plone sharing mechanism.
• Site administrators configure the fields available to users in the AutoUserMakerPASPlugin ZMI page.
• The ShibbolethPermissions ZMI page shows all existing grants.
  • The administrator can delete them on this page.
  • This page has links to each page that has user sharing; edit them there.
• Users grant permissions by clicking on the “sharing” tab of any page that they have permissions on.
  • This will probably require some user education.
  • Fields are regular expressions, which includes simple strings.
  • Some fields can have complex values like LDAP distinguishing names.
  • Fields that don’t have values match everybody.
  • There is no simple *match everybody except* syntax.
This is the top of the sharing tab. Scroll down to get to the Shibboleth Permissions section.
The titles come from the AutoUserMakerPASPlugin configuration.

Adding a grant to a specific user. A User ID of just abrenner would match abrenner@ithaka.org too.
Since we've set up a permission, we've now got an edit area.
ZMI: ShibbolethPermissions

The path is a link to the page that the user has shared.

The attribute list shows those items the user filled in, with the actual HTTP header names.
ShibbolethPermissions Python

- ShibbolethPermissions/Extensions/Install.py
  - Instantiates an instance in the acl_users folder
  - Gets the default folder_localrole_form, inserts the TAL based Shibboleth forms, and saves the updated form in portal_skins/custom
  - Installs several small functions that get called by the forms to get and set the configuration

- ShibbolethPermissions/permissions.py
  - Defines the ShibbolethPermissions class
    - Handles the storage of the user’s permission sharing (CRUD)
  - Defines the ShibbolethPermissionsHandler class
    - Handles the ZMI forms
The software is available at:

- [http://www.python.org/](http://www.python.org/)
- [http://www.zope.org/](http://www.zope.org/)
- [http://plone.org/](http://plone.org/)
- The Plone site has complete python/zope/plone installers for Windows, OS X and SuSE Linux, with a “Unified Installer” for other Linuxes, BSDs and Solaris.
- Installing by hand is just the common `./configure; make; make install` process for Zope, running mkzopeinstance to create a directory tree and unzipping Plone in it.

- [http://tid.ithaka.org/software/](http://tid.ithaka.org/software/)
  - AutoUserMakerPASPlugin
  - ShibbolethLogin
  - ShibbolethPermissions